International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-7, Issue-6S5, April 2019

Methods and Trends in Natural Language Processing Applications in Big Data

Joseph M. De Guia, Madhavi Devaraj

Abstract: Understanding the natural language of humans by processing them in the computer makes sense for applications in solving practical problems. These applications are present in our systems that automates some and even most of the human tasks performed by the computer algorithms.The "big data" deals with NLP techniques and applications that sifts through each word, phrase, sentence, paragraphs, symbols, images, speech, utterances with its meanings, relationships, and translations that can be processed and accessible to computer applications by humans. Through these automated tasks, NLPs achieved the goal of analyzing, understanding, and generating the natural language of human using a computer application and with the help of classic and advanced machine learning techniques. This paper is a survey of the published literature in NLP and its uses to big data. This also presents a review of the NLP applications and learning techniques used in some of the classic and state-of-the art implementations.

Index Terms: natural language processing, deep learning, artificial neural networks, big data applications.

I. INTRODUCTION

The technology of natural language processing (NLP) is another field of computer science that deals with the human language and allowing the computer or machine to understand and process using algorithms. The evolution of NLP since its inception in the early 1930s were using guides, rules, and logic [1]. The birth of NLP advanced through the use of computation and currently data-driven ad learning using statistics, probability, and machine learning [2,3]. In the present information age where all data-driven systems were processed by computers, NLP have leverage the power of solving human-machine interaction by understanding the natural language of humans. The core areas of NLP are language modeling, morphological separation, parsing, and semantics. Currently, "big data" makes it more significant for NLP to be more useful in solving problems in combination with the different learning techniques.

The Internet and its network of hosted repositories containing the "big data" of structured and unstructured data and information are websites, social networks, scientific articles, journals, datasets, etc. These "big data" deals with NLP techniques and applications that sifts through each word, phrase, sentence, paragraphs, symbols, its meaning and translation that can be processed and accessible to computer applications. Some of the tasks available to NLP can be the following: machine translation, generation and understanding of natural language, morphological separation, part of speech tagging, recognition of speech, entities, optical characters, analysis of discourse, sentiment analysis, etc. Through these tasks, NLPs achieved the goal of analyzing, understanding, and generating the natural language of human using a computer application and with the help of learning techniques.

This paper is a survey of the published literature in NLP and its uses to big data. This also presents a review of the NLP applications and learning techniques used in some of the classic and state-of-the art implementations. The paper is divided into 4 sections, section 1: Overview of NLP and its uses in big data; section 2: Taxonomy; section 3: Related Works composed of Sentiment Analysis; Deep Learning NLP applications; and Information Extraction NLP applications.Finally, conclusion and further studies for the next steps is imparted to the next researcher.

II. OVERVIEW NATURAL LANGUAGE PROCESSING AND BIG DATA

The area of natural language processing in the computer science deals with the computation linguistics. This area is also influenced by several learning domains such as statistics, probability, and machine learning. Other influences in social sciences includes psychology, philosophy, cognitive science, and linguistics. Understanding the natural language of humans by processing them in the computer makes sense for applications in solving practical problems. These applications are present in our systems that automates the some and even most of the tasks performed by the computer algorithms. The core areas of NLP are language modeling, morphology processing, parsing, and semantic processing. Some of the applications in NLP includes information extraction, text classification, summarization, question answering, image and video captioning. Language modeling deals with the quantification of words on a given sequence that is used in handwriting recognition, optical character recognition,speech recognition, part-of-speech tagging, parsing, machine translation, and other applications. The morphological process is a means to separating the components of words to adjust its meaning according to syntactic and communication context such as affixation, prefixation, suffixation, etc. Parsing is a process of analyzing a sentence by taking each

Revised Manuscript Received on December 22, 2018.

Joseph M. De Guia, School of Information Technology, Mapua University,Muralla St., Intramuros, Manila Philippines 1002

Madhavi Devaraj, School of Information Technology, Mapua University,Muralla St., Intramuros, Manila Philippines 1002

word and determining its structure from its parts using parser and grammar. The semantic processing is meaning representation of words, phrases, sentences, or the meaning of text. The application that deals with the information extraction from unstructured documents. Thistask which process unstructured texts in documents by using logic and reasoning. The logical context of the input data that can also be used in translation of text to other languages or an interpretation based on category or context of the data. Text classification is classifying text documents into categories or clusters making it easier to manage and sort. Another application is text summarization that deals with the process of summarizing the important key information from documents. Question answering is a task that automatically answer questions by inferring the most probable answer to humans in a natural language. Machine translation task is the use of a translator text or speech to convert them the language to another one. Finally, image and video captioning is the process of generating text description of an image or video. In Fig. 1 the conceptual depiction of NLP how it will become more part of our daily lives through our interaction with the machines and devices with the available applications. In Fig. 2 list all NLP applications available in research, commercial, and state-of-the-art in uses in artificial intelligence and deep learning.

A. Natural Language Processing in Big Data

The growth of data that are too complex to be processed and its conceptual characteristics such as volume, variety, velocity, and veracity in many systems uses advanced tools to reveal meaningful information and insights. Social networks revolutionized the big data where anyone can share, upload, and crowdsource data that can be used in many ways. Facebook as a social network application has 2 billion active users (Facebook, stats 2017).

Figure 1. NLP integrated in big data and AI its impact to human and society [111]

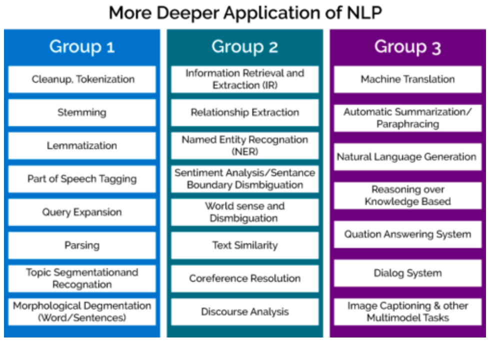

Figure 2. Key Application areas of NLP [112]

Google as a search engine, searches over to 3.5 billion per day (Google trends, 2018). While tech companies such as Amazon, EBay, and others, process millions of queries and transactions with terabytes of data in their enterprise systems. Similarly, in small and large organizations. stored data and information in their enterprise systems are structured and some unstructured such as transactions, records, business processes and even simple notes and emails are in raw form. Many significant problems in the financial, retail, healthcare, academe, government, and in any sector of the society now be solved using big data. Regardless of any data and information that is largely made of human language NLP can be leveraged to access the information at a speed, with accurate and actionable results. It was estimated by IDC in 2020 there will 44 trillion gigabytes of data or generated knowledge can be essential for NLP uses for big data.

The architectures and computing resources in big data were described in Apache Hadoop. This is computing framework in large scale processing of thousands of computing resources. Inside the Apache Hadoop is MapReduce that process and generate big data sets. According to Facebook, the company runs the world's largest Hadoop cluster that is more than 4000 machines with hundreds of millions of gigabytes. The ocean of content on the web being indexed by Google also use MapReduce to manage the large clusters of servers.

To use this framework in NLP, Apache Spark extends the Hadoop framework to leverage the core linguistic applications such as language modeling, parsing, morphology and semantic processing. Some applications of NLP in big data were core applications such as text summarization, sentiment analysis, text classification, information extraction, machine translation, question answering and dialogue systems, and others. An example of NLP uses in big data is the distributed pipelines of linguistic processors. There are available linguistic processors such as IXA pipes and NLP toolkits in big data processing. These linguistic processors help in the annotation to lower the barrier of using the technology with expensive compilation, installation, configuration processes to use tools and toolkits.

III. RELATED WORKS

The following related works are the surveyed papers from the applications of NLP. These NLP applications describes in the next sections will present the methods and implementations thorough the concepts, models, paradigms, and works focused in solving problems using sentiment analysis, deep learning, and information extraction.

A. Sentiment Analysis in NLP

Sentiment analysis or opinion mining is an application of NLP, artificial intelligence, machine learning that determines the contextual emotion or affective states of the subject or person's attitude based on the information, interaction, or event. The sentiment analysis task is to classify the calculated polarity of the person with respect to the given opinion in any form whether its "positive", "negative", or "neutral". In other simple language, the emotional states can be "happy", "sad", "angry", and other emotions that can be described by a person. To determine a sentiment of a person for example when using or buying a product or service, we usually ask the opinion of others, search on the Internet for reviews, comments, feedback regarding the experience or recommendations in the public domain. This helps the person to decide based on the consensus or opinion of others. The Internet is a great tool to scan the general opinion of the public and the sentiments made from different social networking sites and websites. However, it will be difficult for humans to collect these data and information through a simple scanning on the Internet. It is therefore essential to have a sentiment analysis system that will be able to process and automate these reviews and content-generated opinions to identify the sentiments. To classify the opinion or sentiment this should be processed by a system where the text or information are collected such as surveys, comments, blog posts, reviews, or posts in social networking sites. The challenge of sentiment analysis is the approach of extracting the emotion, discovering the objectivity or subjectivity, the use of different features, classification of the sentiment, and the accuracy of the result of the analysis.

a.Taxonomy

Sentiment analysis tools can answer the questions such as "Is the customer satisfied in the product or service being provided?" "What are people tweeting on the current issues affecting them?", "How the reviews or comments of users influence their buying habits?". These questions can be answered by processing the available data in a sentiment analysis system. The sentiment analysis system is a classification method as illustrated in Fig. 3. Another illustration on Fig. 3.1 showing the main classification levels of sentiment analysis having the following: document-level, sentence-level, and aspect-level. The document-level classify the sentiment of the whole document. While the sentence-level classification looks at the expressed subjectivity or objectivity in each of the sentence. The subjectivity expressed explained [13] determines the polarity whether positive or negative opinions were expressed in the sentences. However, there are documents that does not express subjectivity according [14]. In the paper of [15] also explained that the comparison with the document-level and sentence-level classification does not justify the entire opinions of the entity. The aspect-level classify the sentiment according to the specific entity being described as sentiments can be expressed in differently in a same item of the entity. Further,it can be divided into three methods: lexicon-based, machine learning, and hybrid method. The lexicon-based method is divided into dictionary-based and corpus-based approach. The dictionary-based approach looks for collection of opinion words and search it in a dictionary for their meanings. The corpus-based approach uses a precompiled or known collection of opinion words and looks in a corpus to find the words with the context and its orientation. In addition, it uses statistical or semantic methods in classifying the sentiment polarity. The machine learning method can be supervised and unsupervised. The supervised learning method use training data where labeled documents are largely known.

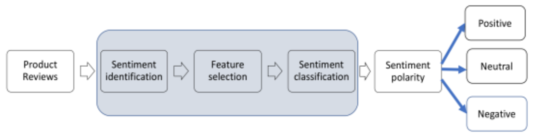

Figure 3. Sentiment analysis system

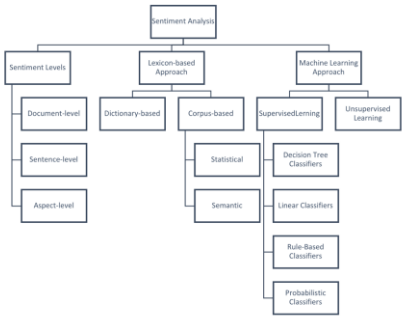

Figure 4 Sentiment taxonomy

The unsupervised learning method relies on the labeled training documents to find the unknown class. The supervised learning method that can be used are the following: decision trees, linear classifiers, rule-base classifiers, probabilistic classifiers. The linear classifiers can be implemented using support vector machines (SVM) and neural networks (NN). While the probabilistic classifiers can be implemented using Naïve Bayes, Bayesian network, and maximum entropy.

b.Sentiment Analysis Processes and Applications

The sentiment analysis process involves the following phases as illustrated in Fig 3.2. The first phase is the data collection is an important step in the sentiment analysis. The appropriate data sets should be determined for the text analysis and classification. In the second phase, text preprocessing reduces the noise of unwanted data by removing the stop words, repeated words, symbols, and others. The third phase is feature selection and extraction that is important to determine the key features needed for the accuracy of the model. Fourth phase is sentiment classification, using different classification algorithms to classify text such as Naïve Bayes and Support Vector Machine (SVM). The fifth phase is polarity detection where the polarity of the sentiment is identified based on the end results – positive, neutral, or negative. Finally, validation and evaluation of the end results to check the accuracy of the sentiment analysis method used. Further discussion is described in the next section.

Figure 5. Sentiment Analysis Processes

Data Collection

Data collection is dependent to the type of data that is defined and supported for analysis and classification of text in the data set. The available data sets in the social networking sites such as Twitter, Facebook, Weibo, Quora, etc. are available for data scraping to collect data from these websites. These networking sites provide an application programming interface (API) to get the data on its the pages. An example is REST API to get the static data such as user information and Streaming API to get and extract the tweets or posted messages. In Facebook, the posts or messages posted on the site were extracted using the Facebook Graph API. While Weibo uses Tencent API that can collect the user posts that can be used for research and review of the influence and network followers.In the paper of [48] used Twitter as a corpus (English) for sentiment and opinion mining. The collected data were subject to linguistic analysis to explain the discovered texts for sentiment classification. The implementation of TreeTagger for POS tagging experiment and claimed that the use of syntactic structures and POS tags are strong indicators of emotional texts. The sentiment classifier using multinomial Naïve Bayes with N-gram and POS-tags feature implementation was efficient and has better performance against previous proposed methods. The next step of their research is to work on the different languages and develop a multilingual sentiment classifier. Another work by [19] used Facebook to identify the opinion and sentiment (English and Malay). Text extracted were clustered into different emotions as lexicons classified into positive (happy), negative (unhappy), and neutral (no emotion). A prototype system was developed to extract data from Facebook and then preprocessed text in a database for sentiment analysis. The text in the database are clustered and determines the emotion by displaying the tagged or classified emotions in a visual format. However, the evaluation of the prototype and the method used were not described in the process. In another work [20] six different data sets using movie reviews, blog posts, and twitter feeds were evaluated using several machine learning (supervised and unsupervised learning) and lexicon based methods for sentiment analysis. The result of their experiment reported that machine learning classifiers are not fit for aspect-level sentiment analysis. The machine learning method is best for document-level sentiment analysis using unsupervised lexicon-based method SentiWordNet [20]. A proposed method for aspect-level sentiment analysis for the movie review have accuracy levels which is similar to document-level approach. It was recommended that this proposed method can be used to tasks processing of movie recommendation systems.

Preprocessing and Feature Extraction

The preprocessing of data acquired from various sources can be reduced the noise and retaining only the essential and significant text before being introduced to full analysis. Some of the preprocessing steps are the removal of the stop words, removal of repeating words, removal of emoticons and symbols, stemming, POS-tagging, feature extraction and representation. The process of breaking the sentence into tokenized parts (words, phrases, symbols) and removing the punctuation marks. The stop words are removed in the preprocessing step. In stemming, the root of the word is determined. While the POS-tagging is to identify the parts-of-speech in the text. Some of the preprocessing tools for sentiment analysis cited in [25] were TweetMotif for tokenization of tweets [5]; POS Tagger for Twitter [5,6]; TweetNLP [7]; Lancaster stemming algorithm [8]; GNU Apsell spell checker [9]; Snowball English stemmer [10]; Stanford Log-linear POS-tagger [11]; andTweeboParser tweet dependency parser [12].

Preprocessing,Feature Extraction, and Evaluation

The research made by [16] use the simple approach of classifying polarity from positive to negative in product reviews. In [13,17] use the movie review website and applied different schemes of getting the polarity values with the prediction of star-rating on a 3-4 scale. Another in-depth analysis of sentiment analysis using prediction ratings on restaurant reviews was studied by [18]. Another approach was made by [19] combined the lexical, knowledge, and learning methods in the area of sentiment in text. In [20] evaluated the performance of various machine learning methods and lexicon method in sentiment analysis of texts to compute the sentiment polarity. A heuristic-based approach was also described for aspect-level sentiment analysis using movies dataset. Survey papers in sentiment analysis and classification discussing the classic and state-of-the art were described by [21,22] sentiment analysis and opinion mining techniques; sentiment analysis and algorithms [23];sentiment analysis challenges [24]; sentiment analysis task, approaches, and challenges [25]; comparative study on different approaches [26].

.B. Deep Learning in NLP

Deep learning models are making advances in the artificial neural network (ANN) applications in computer vision and pattern recognition. The recent researches in NLP utilize the new deep learning approaches. Using deep learning models in solving NLP problems have the attention of most researchers and presents state-of-the-art models, performance, accuracy, and superior results to NLP processes. Deep learning models presents network layers where input data learns and transform in a neuron-like signal with computed vector component values in the output. The ANN model is like thehuman brain where the nodes are interconnected (like neurons) and receiving inputs and transmitting signals (synapse). The neural network method solves problems which similar to human brain. This architecture was implemented in many tasks of the NLP applications such as speech recognition, machine translation, speech recognition, image recognition, and others.

a. Neural Networks and Deep Learning

The ANN are networks of networks designed like how the brain activity works and employs the network of connected nodes (neurons) to performs a specific task through its learning ability. The neuron which receive input through an activation change the state generates a type of output transpired in the process. This is process flow of neuron signal is shown in Fig 3.3 An artificial neuron simulates the biophysical neuron with inputs and outputs (Wikipedia, ANN). The neural net forms a network and the output of the neurons to the other network of neurons forming a network graph. The network graph itself through the activation in each node in the network and its function can be computed to produce the learning through the learning-rule [32]. The components of an ANN are the neurons, having an activation, threshold, and activation function. An input neuron has no origin and serves as an input into the network while the output neuron has no successor and serves as an output of the whole network. The ANN are connected by a directed line and each connection has a weight. If the weight has bias on it this is added to the sum of the inputs and will be the given threshold for the activation function. A learning rule is an algorithm to look at the parameters of the neural nets to achieve the desired the output. This learning algorithm rule change the weights and threshold of the variable in the ANN. The network of artificial neurons similar to the brain where each neuron transmits signal to another neuron in the process of synapse. In this process, neural network model is designed to have layers of nodes to perform the input transformation and the process of learning takes place. The input layer receives the signal from the nodes and the last output layer performs the computation function of the summation. The nodes in the output layer computes the values it receives from the input values and learns when showing large numbers of input and output pairs. The errors and losses made in the network can be corrected in the output node where the individual weights are adjusted based on the computed derivatives. This implementation is using stochastic gradient descent. There are different network layer orientations as designed by researchers in this area. This is to distinguish the types of networks where every node in the layer are connected and the number of layers itself. The most basic neural network is having a sequential layer known as feedforward neural nets (FNN). This basic neural net does not form a cycle and was the simplest type of ANN. Another network is the Deep neural nets (DNN) having multiple hidden layers and was termed deep with its many layers. Fig.3.3 shows the different neural network as described. In the next section the types of ANN will be described.

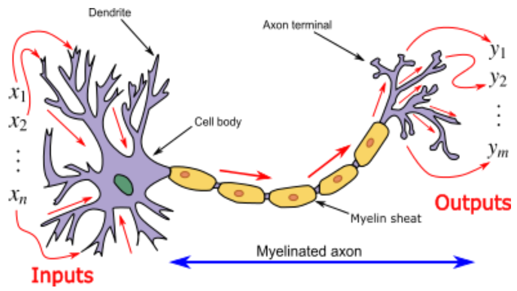

Figure 6 Biological process of Neuron and its layers and processes [113]

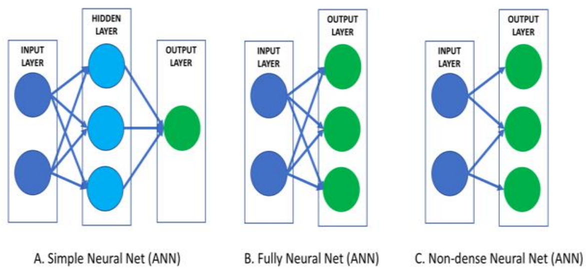

Figure 7 An illustrated example of simple Artificial Neural Nets (ANN)

b. Taxonomy

Deep learning method is based on learning representation and feature extraction. Several research experts who made its resurgence in 2006 with the works of [27, 28, 29]. The papers presented in deep belief networks challenged the various state-of-the-art implementation using vast availability of data sets in computer vision and automatic speech recognition. In classical machine learning supervised and unsupervised these are task-oriented using training data and labels for unknown data using features. In the last decade, deep learning contributed extensive applications in image recognition [28], speech recognition [27], natural language processing [52,53]. Deep learning methods [36,41] was extensively explored and have been proposed in several researches and classified into Restricted Boltzmann Machine, Deep Autoencoder, Sparse Coding, Convolutional Neural Network and Recurrent Neural Networks as presented in Fig. 3.5.

c. Convolutional Neural Networks

The CNN is a class of deep learning having one or more convolutional layers with fully connected layers. The hidden layers of CNN have convolution layers, pooling layers, fully connected layers, and normalization layers. Figure 3.6 illustrates the CCN architecture. It was based on the Fukushima's convolutional architecture (1980) named as "neocognitron" [34]. The term itself derived from a mathematical operation "convolution" where two functions produces another third function that shows the shape of the function is altered by the other. The convolution operation takes place in the in the input layer and passing the result in the next layer. The convolution copies itself to the neuron that is based on theshape, image or visual stimulus. The convolutional neuron has receptive field to process the data being fired to layer looking for the same features. This receptive field is an input area of the neuron. In a CNN layer, the previous layer is larger than the area of the receptive field. However, FNN which is fully connected can classify and learn these features specifically for images. The number of pixels in an image as a variable being used in the convolutional process can have a large number of neurons. This is relevant to the input size in the network layer. If an image has a size of million neurons and has a hundred various features in an area of ten thousand, can this problem reduce the number of parameters? The convolution operation solves this problem by making the network deeper with a number of parameters [38]. It makes sense to train one hundred neurons that can also be used ten thousand times. Therefore, this process whatever size on the training solves the problem of vanishing or exploding gradients in the multi-layer backpropagation. The vector weights and bias in CNN called filter is a feature input of the shape or image being shared all across the receptive fields. CNN uses a class of FNN that needs small preprocessing effort. This model leverages the 2D structure and used as an input. CNN is best suited for image processing using 2D images as an input. Similarly, it also works in speech recognition applications. Another design feature of CNN is pooling. This design feature combines the neuron outputs into clusters in each layer of the network – using max pooling for maximum value of clusters and average pooling for average clusters [39, 40, 42]. CNNs are largely implemented in image and video processing. Likewise, it is also used in the speech and natural language processing [41,42,43,44]. In Fig. 3.5 a CNN model is illustrated. The networks have unique filters in each of the input layer, with receptive field sizes accordingly. Max-pooling is also illustrated with k=2 as a pool size of 3.

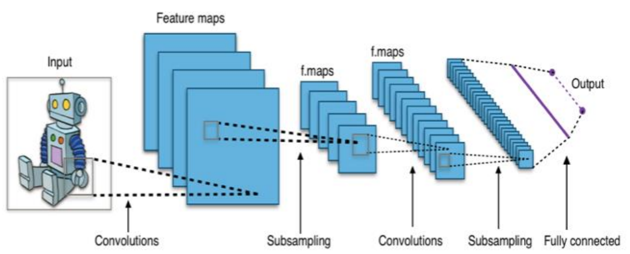

Figure 8 Convolutional Neural Nets Architecture [114]

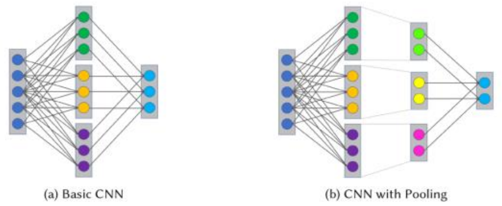

Figure 9 An illustrated example of Convolutional Neural Nets (CNN) [4]

d. Recursive Neural Networks

A Recursive Neural Network (RvNN) is a variant of ANN that use a set of weights recursively shared between layers to minimize training. The model allows over structured input to predict a variable-input or scalar value in the given traversal path order of a model sequences. The natural recursive structure of RvNNs where words and phrases are combined to form a hierarchy can be described or visualized in a parsed tree. The tree-structure use syntax and meanings of the sentence structure [47]. RvNN are popular in NLP tasks as it allows to model a parse tree structure and learning sequences. A singe tensor of weights can be used at the low-level tree and at successive high-level tree structures. The nodes in RvNN are dependent on the results of the previous nodes and look for the feedback of the node. The non-terminal node in the parse-tree is determined by the leaf or child generated. RvNNs are used in word embedding, logical terms [46] and other models and frameworks.

e. Recurrent Neural Networks

A Recurrent Neural Network (RNN) is a variant of ANN to model sequential nodes where the connected layers form a directed graph. This model incorporates a temporal layer, a behavior for a sequential information or a simple type that can be finite and infinite impulse. RNN has a hidden layer that constantly changes that updates itself and the status of the network as illustrated in Fig. 3.8 and Fig. 3.9 the networks (a) basic RNN and (c) unrolled RNN which are similar (b) stacked RNN – sequential RNNs (d) unrolled RNN stack. While (e) bidirectional RNN use two RNNs with different directions and combines the output such as a reverse sequence of data. The nodes have a varying-temporal values and activation values. Similarly, the connection has modified weight in the input nodes, output nodes, and hidden nodes.

RNNs nodes perform the same tasks in the sequence where the output layers are dependent on the other previous layers results. RNNs nodes can remember the previous results and use this information in the process. With this process, RNN is best fit for many NLP tasks as cited in [87] such as language modeling [88, 89, 90], machine translation [91,92,93], speech recognition [44, 45, 94, 95, 96, 97], image captioning [98], handwriting recognition [51]. There are different RNNs such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) integrate varieties of gates and memory cells to capture temporal activity sequence [51]. Long Short-Term Memory incorporated memory cell to store contextual information, thereby control flow of information into the network [44].

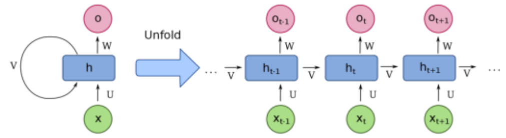

Figure 10 An illustrated example of Basic RNN [115]

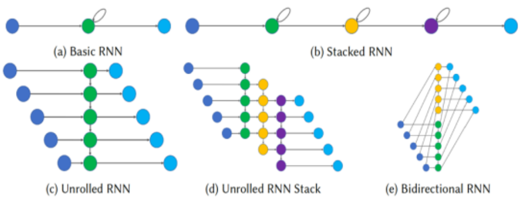

Figure 11 An illustrated type of RNN [4]

f. Long Short-Term Memory Networks

The LSTM is a sub unit of RNN. An LSTM network have a cell, an input gate, an output gate and a forget gate seeFig 3.10 LSTM network. The cell recognizes the values over assigned the time periods. These gates control the information flow into and out of the cell. The gates calculate the missing node state by from the rest of the gates. In LSTMs, it has several individual neurons connected in a manner designed to retain specific information. The LSTM allows the mistake of the node to go back having an unlimited number of pass called backpropagation. Using the LSTM blocks in RNNs to form LSTM networks, the information can be retained almost indefinitely while irrelevant information can be forgotten. However, if information is disregarded, it cannot be recovered, even if its presence is desired later.

Some of the current research explores to correct this problem by retaining larger amounts of data and simply changing which parts of particular examples are attentive [101, 102]. Another work made by [103, 104] exploring the Gated Recurrent Unit (GRU), perform as well as or better than standard LSTMs in many NLP tasks. This includes handwriting recognition [43, 49, 50] and natural language generation [51], language modeling [105] and machine translation [53], acoustic modeling of speech [54], speech synthesis [55], protein secondary structure prediction [56], analysis of audio [57], and video captioning [58] and others.

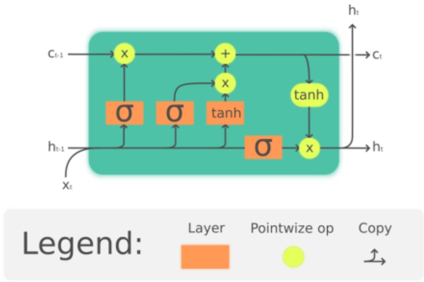

Figure 12 y Networks

g. Autoencoders

Autoencoders is a variant of ANN for data encoding in machine learning methods. The autoencoder needs to learn an encoding or representation to reduce the dimensionality of the data. Autoencoders has been applied in several researches for generative models and notably the most dominant implementation in the deep neural networks has stacked autoencoders [59]. The purpose of an autoencoder is compress the data from the input layer and then looking for a match in the network to uncompressed the code on its original form. This autoencoding process reduces the dimensionality of the code. In some network, the autoencoders are stacked together in layers for image recognition. The features are encoded and learned within the network until the code that matches the image emerged in the output layer. Another approach is the use of generative model that feeds the known image and then generate the desired output by learning the codes itself without knowing the codes [60, 61]. The design of the auto encoder is feed forward network, multi-layer perception, and non-recurrent neural networks that are unsupervised learning models. An auto encoder has encoder and decoder parts of the network and its tasks in the network. In Fig. 3.10 the autoencoder has 3 fully connected hidden layers where the hidden layer takes the code, latent variables, latent representation in the decoder stage maps the code with the activation function and the weight (bias vector) to reconstruct the shape corresponding to the design of the autoencoder. In Fig. 3.11 Encoder-Decoder model RNN variant of autoencoders (a) showing the attention method (b) determining the encoding in the output layer from the decoder. The autoencoder variations based on its design and techniques are Denoising autoencoder, Sparse autoencoder, Variationalautoencoder, and Contractive autoencoder. The autoencoder variants are presented in the next sections.

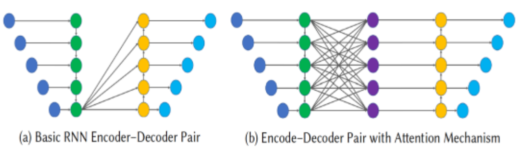

Figure 13 An illustrated sample Autoencoder [4]

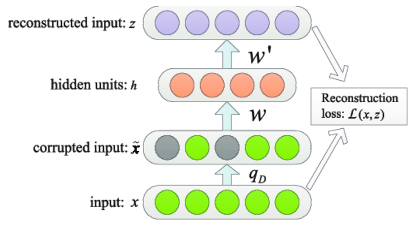

Figure 14 An illustrated sample of DenoisingAutoencoder [63]

Denoising Autoencoders

The denoising autoencoder (DA) was presented in [62]. In this task, the feature that is being learnedto come from an incomplete data or corrupted sample input data. In the DA task, it reconstructs the input data by making the values of the data to zero by using the stochastic mapping method. The DA is initialized into several layers of training. The network is trained to generate the next higher-level features of the layer.

The training ensures that the autoencoder network captures robust properties of the input data distributions. Furthermore, DA can be stacked to learn the useful features of incomplete input data which has less classification error. In Fig. 3.13 it shows an illustration of the DA.

Sparse Autoencoders

In a sparse autoencoder (SA) design there are many hidden layers that that of the input layer. However, the many hidden layers only have a few that are active at the same time. The sparsity is being done by the addition of terms in the training or forcing the terms in the loss function or making the hidden activation layers close to zero which uses k-sparse autoencoder method [64]. The use of sparsity term makes the model to learn feature representation that are robust, linearly separable, resistant to changes in learning. The SA model is very good in getting low dimensional features from high dimensional, compact, and complex input data using supervised learning. SAs are recommended for tasks in image and video recognition which has high dimensional data. The basic design of a SA is illustrated in Fig. 3.14 where it consists of a single hidden layer, h, that is connected to the input vector x by a weight matrix W of encoding step. The hidden layer then outputs to a reconstruction vector x˜, using a tied weight matrix WT to form the decoder. The activation function is f and b are the term bias.

VariationalAutoencoders

The variational autoencoder (VAE) is a similar model of a classic autoencoder with high distribution of its latent variables. The basic design of VAE is consisting of encoder, decoder, and a loss function. The first layer of VAE is the encoder that takes its input and covert the code to a latent vector. Like a standard autoencoder the latent vector (an image) uses a generative model that add constraints to the encoding network. This is illustrated in Fig. 3.15 VAE uses variational method for latent representation learning or stochastic gradient variational bayes (SGVB). This method has an additional loss component and use SGVB algorithm where the parameters of the learning model, recognition and generative generates the data from a directed graphical model and the encoder uses approximation function of the model. In deep generative models, researchers are exploiting this method to generate near-to-real images. In NLP application, the method of deep generative models to text data generate realistic sentences from the latent code space. Using VAEs in sentence representation learning, the model consists of an encoder and generative model network that encode data to latent representation and generate samples from the latent space. A paper published by [106] proposed an RNN VAE model that is illustrated in Fig 13.5 RNN-VAE model for sentence generation as cited in [87]. This model has distributed latent representation of an entire sentence and explicit global sentence representation. Another proposed model by [107] copied the latent code and a set of structured variable looks at the semantic features of the sentences. This model generate reasonable sentences structures classifying tenses and sentiment attributes.

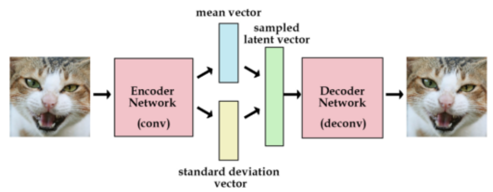

Figure 15 An illustrated sample of Simple Variational Auto encoder [118]

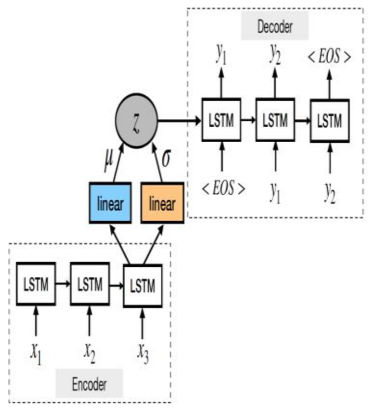

Figure 3.15 An illustrated sample of RNN-VAE model for sentence generation [106]

Contractive Autoencoders

Contractive encoders make learned representation resistant to slight variations of training values. The model introduces the penalty term to the cost function for efficient representation. The cost function is the mean squared error that corresponds to the Frobenius norm of Jacobian matrix of the encoder activations in relation to the input. The partial derivative of the of the neuron's activation value with respect to the input value to sense the Jacobian value for the penalty. The formula of sparse autoencoder are similar to contractive mapping where the low activation values of the neurons on sparse autoencoders are flat because of the small Jacobian matrix value. Contractive autoencoders are also similar to denoising autoencoder where both are invariant to changes and distortion. While the denoising autoencoder favors reconstruction and similar to encoder function for contractive autoencoders.

C. Information Extraction

Information are available in any systems that store the tacit and implicit knowledge. These are contained in many documents and other available file formats that are stored in repositories that are created and maintained in a computer, enterprise system or the Internet. To be able to reuse the information and disseminate, it is important to extract the knowledge contained within these document repositories. The task needed in order to improve the way we make decisions and improve operational efficiency is to automate the process of getting away from the repetitive and menial work of extracting information and knowledge into the system. The task of information extraction (IE) system is to get the data of interest and its relation from a document and produce a structured output like a relational table and stored in a database. Simple and statistical methods were implemented in processing texts using NLP tasks such as classification, pattern recognition, grammar and rule-based methods. IE is an essential field in text processing that involves NLP, computational linguistics, and text mining. IE is mainly the essential part of knowledge management systems. Currently, IE applications are present in image and video annotation processing, speech processing as well as content extraction. The present information extraction systems use machine learning. The extracted information described are named entities, relationships, and events.

Information Extraction in NLP

The task of extracting explicit or implicit information from text through the use of an algorithm is known as IE. The IE tasks is to get the data and its relations from a document and produce a structured output like a relational table and stored in a database. Simple and statistical methods were implemented in processing texts using NLP tasks such as classification, pattern recognition, grammar and rule-based methods. IE is an essential field in text processing that involves NLP, computational linguistics, and text mining. Currently, IE applications are present in image and video annotation processing, speech processing as well as content extraction. The field of IE deals with the task of automating the text management in the system through its processing, storage, and transmission. The task of classifying and indexing large volume of documents have been automated by information retrieval (IR). In processing and modeling natural language, NLP tasks have accomplished and solved many of these problems. In IE, most of the users are interested to extract the following according to the existence of any set of documents that has: (1) named entities, (2) relations (3) events.

An example of IE system in Fig. 3.16 presents the workflow for processing a known document and indexed using a template. The method using a template detection and index data extraction and tasks for extraction rules using a training are applied to the extraction documents where the extracted index data is used for feedback into the system [69]
.

Information Extraction Systems Architecture

Every IE system performs the defined task according to the design paradigms. There are two approaches in building IE systems according to design paradigms: Knowledge Engineering and Automatic Training.

Information Extraction Systems Design Approach

The Knowledge Engineering design is based on the knowledge of person whose familiar or an expert of using the system or the "knowledge engineer". This is also the same for a system developer whose familiar with the requirements, design, and the functionality of the system. The design of the system is based on the requirements how the user would like to use based on the business rules - extracting of relevant information, using a corpus, and the desired output. The development of the IE system is an iterative process. The set of rules are written, tested on a corpus of text for training, and evaluating the results of the output according to its accuracy and performance for fine tuning the entire IE system.

The Automatic Training approach doesn't need an expert, system designer, or engineer who knows the entire IE system. This approach only need a person who can annotate the text properly based on the corpus of texts and the information being extracted. The requirement for this approach is to annotate and train the texts appropriately to obtain the knowledge from the system.

Information Extraction Systems Modules

From the design paradigms described, the IE system architecture can be built according to the core elements of the system. The core modules of an IE system are tokenizer, lexical and morphological processing, syntactic analysis, and domain-specific analysis. Additional modules can be added to the core depending on the required task or need by the users. These are word segmentation, parts-of-speech tagging, coreference, and merging partial results. See Fig. 3.17 for the illustration of the IE system architecture [70].

Tokenization

Tokenization module task is to separate the input text into sentences and tokens. Languages have different ways how words and sentences can be separated or based on orthographic properties where each word boundary can be identified. Such examples of trivial problems are for European with word borders, Chinese and Japanese for its orthography word boundaries. It is recommended to have a word segmentation module added into the IE system.

Morphological and Lexical Processing

Morphology is the study of how word structures are being formed and its relationship to other word. The morphological task is simple for the inflectional morphology in English that only list the variants of the word of interest. For other languages, that has a complex morphology the need for morphological analysis module is required.The lexical processing of the token from the tokenization module looks for lexicon or features. The essential part of the process and analysis is the identification of names (name recognition) that can be identified by using rules, training of the annotated corpus. The name recognition uses the assigned lexical features (lexicon lookup, parts-of-speech tagger, automatic taggers) required in the processing. Parts of speech tagging is essential part the analysis in this module. This eliminates the ambiguity in some languages to avoid error and the word inflection. The process makes a filter for good system with accuracy and efficient performance

Parsing

Parsing is a grammatical analysis that checks for the grammar of the sentence in a document. In IE systems, it is only looking for texts specified important and relevant to the task. There can many parses in a sentence. However, the syntactic properties and analysis can be essential or some parts of it can be ignored.

Coreference

The coreference module handles the problems in: naming alias or coreferring, pronouns and their antecedents, definite description, and ontological information for domain relevant entities, temporal events, etc. These tasks are finding the links from previously extracted named entities. The process also recognizes the referential relations in the expressions.

Domain Specific Analysis

This module is the core of the IE systems. This module prepares the IE output using templates or attribute-value pairs. The text fills the field with the value or pointers to other objects. The template designed is based on the required objects that use extraction patterns or rules. This can be generated manually or using automatic learning methods. For the domains requiring specific pattern rules, atomic or molecular method can be used. In molecular method, it matches all arguments starting with high reliable rules to capture the common core relevant patterns. If the system has more problems on it can generate rules that starts with high precision and low recall scores or the opposite low precision and high recall. The atomic method has a domain module with arguments in combination to the template instead of the syntactic relationships. It is assumed that domain-relevant events for any name entity recognition can have high recall and low precision.

Merging Partial Results

Merging Partial Results looks after the all the information that is combined before the final template and generate the result. The IE system use the merging module with an algorithm deciding which templates can be merged or combined. Depending on the desired outcome of the IE system, the Knowledge engineering method performs data analysis with predefined rules and merging. While the automatic training method is use training rules to merging tasks.

Taxonomy

The taxonomy of IE systems related to the structure of data and the information that is extracted to data dimensions. IE tasks can be suited for specific template formatted documents such as HTML or XML. IE can also be used to extract relevant information without training, preprocess or classify, and structure semantic data. IE can automatically extract sequence of text. In Fig. 3.18 illustrates the taxonomy of IE for different classification of data (1) structured data, (2) semi-structured data, (3) unstructured data. Another IE taxonomy that illustrates the structured view of information where the task and sub tasks is illustrated in Fig 3.19. illustrated IE taxonomy.

Structured Data

Structured data is a relation data where the structure or schema has assigned data making the information relevant to the assigned named entity and relationship between entities. Named entity relationship (NER) is used to extract names of persons, organizations, or any miscellaneous text. Relationship extraction is the identification and extraction of relationship automatically between two or more named entity.

Semi-Structured Data

The extraction can be made without any analysis effort required. There is no semantic processing or analysis needed. Similarly, no syntactic processing and morphology needed. This can be XML-encoded or HTML structured pages.

Unstructured Data

Unstructured data are pure plain text. Processing unstructured data requires linguistic processing to extract the knowledge and NLP tasks to deploy rules to fill a database.

Semantic-based

The Internet has googol of information and generated in textual and other media format which differs to one another in semantic level. The semantics types are Lexical Semantics, Statistical Semantics, Structural Semantics, and Prototype Semantics [108]. Scraping information from websites can be done using the methods of presentation regularities and domain knowledge. The author explained that the website should be divided into segments of information blocks to organize the content into several divisional groups. In this process of partitioning the website content the trade-off was loss of information.

XML-based

There are many challenges in information extraction from a semi-structured document such as a website without design, format and semantic heterogeneity. To solve these challenges, an Extensible Mark-up Language (XML) was proposed to organize the web data. XML is a system based on the field structure of the individual page [109]. The fields of the same record contain same of that website. Other fields in the website may be located in other pages and can be linked based on the reference.

Ontology-based

Ontology is a structure of objects, its properties and relations between the contents or objects. This is specified for a specific domain where the tasks are formal and explicit according to ontology process. Application of ontology-based can be used in processing information contained in text, semantics in the Web, and evaluating the quality of the ontology systems. The ontology-based system process unstructured and semi-structured text by using an ontology guide to extract the information types and generate the ontologies of the output.

Information Extraction Systems

The IE system can be divided into manual and automatic learning method. The following systems are the existing system based on the taxonomy provided.

Manual Information Extraction Systems

The IE systems from the Message Understanding Conference (MUC) [72] dates back from 1987 produced several IE systems. The following systems are: FASTUS, a cascaded non-deterministic finite automaton; GE NLTOOLSET knowledge-based, domain-independent processing tool; PLUM, manually generated rules.

FASTUS

The Finite State Automaton Text Understanding System (FASTUS) [73] is a cascaded, nondeterministic finite state automaton (non-DFA) FASTUS was used in the MUC-4 with the theme of terrorism in Latin America from several news articles. The system has the following processing steps, (1) triggering, (2) recognizing phrases, (3) recognizing patterns, (4) merging of incidents. The first step of process is to search the trigger words such as person names in every sentence. In the second step, nouns, verbs, and word classes are identified by the non-DFA. In the third step, no parts-of-speech (POS) tagger was used. Instead the patterns are identified and hard-coded to extract incidents detected, the last step incidents were merged. This system is relatively small with a very large dictionary in the process. The rules developed performed very well and effective [73].

GE LTOOLSET

The GE NLTOOLSET [74] is a knowledge-based, domain-independent core of text processing tool. This tool has three processing toolset steps identified as: (1) pre-processing, (2) linguistic analysis, and (3) post-processing. In the pre-processing step, segmented and irrelevant parts of the text are removed. Then this process identifies the phrases that are template activators and parts of the text are marked with discrete events. In the linguistic analysis phase, parsing and semantic interpretation are being inspected. The last phase is the post-processing that picks the templates and maps semantic categories and roles. The system was populated with the knowledge base of a core-lexicon feature and functional grammar [74].

PLUM

Probabilistic Language Understanding Model (PLUM) [75] was introduced in MUC-4. The system has a preprocessing, morphological analysis, parsing, semantic interpreter, discourse processing, and a template generation modules [75]. In the preprocessing module, the boundaries of the message were identified along with the header, paragraph, and sentence. The next module is the morphological analysis where the POS tagging with trained models for recognizing Spanish and English words were tagged and identified. In the parsing phase, one or more non-overlapping fragments of the input sentence were generated. Then the semantic interpreter processes the fragments from the previous module. The semantic interpreter, lexical semantics and rules were constructed. The discourse module builds the event objects with the relevant events in the message. The template generator with the built structures by the last phase generate the templates which is the final output of the system.

Automatic Information Extraction Systems

The slow and complex manual IE systems devised by system developers and engineers in the MUCs led to the automation of this task. Machine learning methods were used in order to automate the IE tasks. Supervised learning, large set of training data is required to learn the rules of the task. Unsupervised learning, a small set of seed rules and annotated corpus with bootstrapping were set-up into the system. These systems are Autoslog, text extraction rule [75]; PALKA, induction method using a frame-phrasal patterns [76]; CRYSTAL creating extraction rules [78]; GATE, a developer environment for NLP [71]; and LIEP pattern generalization [79].

Supervised Learning Information Extraction Systems

Supervised learning method makes use of training data to generate extraction rules. In this method, large training dataset has to be annotated in order to extract the information of interest. The main issue of supervised learning IE systems is the preparation and availability of the training dataset. Most IE systems needs big dataset of annotated documents for the extraction task.

AutoSlog

The first system to learn text extraction rules from a training dataset was AutoSlog [76]. This system extracts information from a text containing domain-specific dictionary of concept nodes. The concept nodes are rules that has the trigger word or words with semantic constraints. It uses a semantic tagger and constraints in the extraction rule. In addition, it does not merge similar concept nodes and handles only free text.

PALKA

This system generates the extraction rule frame and phrasal pattern which was called Frame-Phrasal pattern structure (FP-structure). PALKA creates a rule and tries to generalize it with existing ones to include a new positive instance into the process. It specializes existing rules to avoid a negative instance if it generates the wrong interpretation when applying existing rules [76].

CRYSTAL

The CRYSTAL system syntactic parser for processing text and annotated training documents. The extraction rules in the training data are created in CRYSTAL to find the most similar pair or rules. The rules are merges together by finding the most restrictive constraints from both rules [78].

GATE

Information extraction system commonly seen on systems is the text based index term method. It uses predefined rules and automatically generate from tagged or annotated training documents. The rule-based systems make the system or knowledge engineer to define the rules to find the index terms. The General Architecture for Text

Engineering (GATE) [71] is a model and developer environment is a free IE system that can be utilized to engineer the development and deployment of simple and extendable modules for text annotation to training data for NLP algorithms. The process of annotation can be made manually or by adding the new annotations and processing over to the corpus. GATE is a complete package for NLP application.

LIEP

The LIEP system learns multi-slot rules similar to AutoSlog, extraction. The events are identified by users under the assumption that a no large annotated training corpus is available. The training sentence, entities of interest are identified in LIEP and choose extraction patterns that maximize the number of extractions of positive examples and minimize spurious extractions. If there is no known pattern to be matched for new example, else it generalizes. If the generalization is not possible or the resulting pattern decreases the quality, a new pattern is constructed [79].

Unsupervised Learning Information Extraction Systems

Unsupervised learning IE systems eliminates the problem of having a statement of the required information. No extraction patterns or training data available to the user. The main challenge is the need to generate the extraction patterns. The extraction patterns can be bootstrapped and expand the initial set afterwards. The existing systems are: AutoSlog-TS [76]; EXDISCO uses a mutual bootstrapping strategy [82]; Snowball [83]; QDIE framework [85].

The development of unsupervised IE systems came to realize because of the expensive cost of preparing the annotated training datasets for supervised method.

AutoSlog-TS

AutoSlog-TS is an extension of AutoSlog that has pre-classified training corpus relevant to each task. AutoSlog allows several rules to be generated. The pattern is needed is to be reliable for the domain. In a next stage of the patterns an evaluation process takes place. The extraction patterns are ranked based on the relevance statistics of the corpus in this phase[76].

EXDISCO

EXDISCO [82] is a mutual bootstrapping method of the availability of patterns and finding the relevant documents. The candidate patterns are generated from the clauses in the documents and ranked in correlation with the relevant documents. The highest pattern is added to the pattern set and each document is re-ranked using the newly obtained pattern set [82].

SNOWBALL

The Snowball [83] is a system is based on Dual Iterative Pattern Expansion (DIPRE) algorithm. DIPRE [84] training data works well with two distinct features, each of which can independently distinguish the class of instances from the other.

QDIE

The Query-Driven Information Extraction (QDIE) model [85] minimize human intervention by through an automation of a set of keywords as input. A pattern is more relevant the more it appears in the relevant document set and less across the entire document set. QDIE uses the Subtree model, a generalization of the Predicate-Argument model [82] and the Chain model [86]. such that any subtree of a dependency tree in a source sentence can be regarded as an extraction pattern candidate.

IV. SUMMARY AND CONCLUSIONS

The technology of natural language processing (NLP) is another field of computer science that deals with the human language and allowing the computer or machine to understand and process using algorithms. The Internet and its network of hosted repositories containing the “big data” of structured and unstructured data and information are websites, social networks, scientific articles and journals, etc. These “big data” deals with NLP techniques and applications that sifts through each word, phrase, sentence, paragraphs, symbols, its meaning and translation that can be processed and accessible to computer applications. The growth of data that are too complex to be processed and its conceptual characteristics such as volume, variety, velocity, and veracity in many systems uses advanced tools to reveal meaningful information and insights.

The second section presents the natural language processing application to sentiment analysis. The discussion about the taxonomy presents the following levels in document, sentence, and aspect level specific to text and the individual items or entity that describes the sentiment classification. Another method by using lexical, machine learning, and hybrid approaches. The lexical-based method uses dictionary and corpus-based approach. The machine learning method uses supervised and unsupervised learning approaches. The supervised learning method use training data where labeled documents are largely known. The unsupervised learning method relies on the labeled training documents to find the unknown class. The supervised learning method that can be used are the following: decision trees, linear classifiers, rule-base classifiers, probabilistic classifiers. The linear classifiers can be implemented using support vector machines (SVM) and neural networks (NN). While the probabilistic classifiers can be implemented using Naïve Bayes, Bayesian network, and maximum entropy. The sentiment analysis processes described the whole steps of how sentiment classification and detection were used. These steps in the process are important to perform the various methods and techniques in sentiment analysis. The surveyed papers presented various sentiment and opinion analysis in-depth knowledge on how NLP was applied in solving sentiment analysis problems. Data sources and tools available for sentiment analysis were exploited by some researchers to compare, analyze, evaluate the performance and accuracy of several experiments and enhancement of the approaches in sentiment classification and validation of polarity detection using different schemes.

The third section described the plethora of applications in NLP are described in this chapter and the uses of deep learning in each of these applications has a great advantage to be explored and discovered. Some of the NLP applications that will greatly affect most of the users is the sentiment analysis where becoming more main stream and popular to extract the real score of polarity in every aspect of the available information on the Internet. Deep learning becomes the new state-of-the-art in solving NLP problems where computation of data is now readily available and can be harnessed with little or less effort using various deep learning models. Current applications of DNNs in NLP are in social media analysis, prediction of the all sorts in the market, government, politics, society challenges, and business processes automation. Research made in this area has the power to change some of the rudimentary everyday tasks of people, businesses, and the society. The proof-of-concepts are present in the researches made by academia, adopted and expanded by industry specifically the Internet and computer companies (Google, Apple, Facebook, Microsoft, Amazon, etc.). The future of NLP applications using deep learning will surely surge in the years to come.

The fourth section in Information Extraction has been essential tasks of most of the applications in NLP, text mining, text classification, and computational linguistics. In this survey paper, it presented the role of IE in structured and semi-structured data that requires NLP methods. Machine learning tasks can be deployed to generate extraction rules that requires large training sets of data. This method automates the cumbersome manual pattern discovery and rule generation. However, this training cost for annotating the datasets open a challenge of automating the process through unsupervised learning techniques. Some of supervised and unsupervised IE systems were presented. Although some of these systems are not the state-of-the-art, the IE field is open for new discovery and might take to revisit the previous system to unleash the new task for the future of this domain. This survey paper is not exhaustive and it is recommended to further look at other published surveys for IE that will support and add value to this research.

Finally, this survey paper is not comprehensive and exhaustive of all the published papers in NLP. It is recommended to next researcher to expand the discussion and have the state-of-the-art or latest trends in NLP applications with the recent performance and accuracy result evaluations. Similarly, it can be another survey for the latest trends in other aspects of NLP.

ACKNOWLEDGEMENTS

This work is supported by Mapua University through the Directed Research for Innovation and Value Enhancement (DRIVE) institutional research initiative.

REFERENCES

1. Hutchins, WJ. (2001). Machine translation over fifty years. Histoire EpistémologieLanguage 23, 1 (2001), 7–31.
2. Jones, K. (1994). Natural language processing: a historical review. In Current Issues in Computational Linguistics: in Honor of Don Walker. Springer, 3–16.
3. Liddy, E. (2001). Natural language processing. (2001).
4. Otter, D., Median, J., Kalita, J (2018). A Survey of the Usages of Deep Learning in Natural Language Processing. arXiv:1807.10854v1 [cs.CL] 27 Jul 2018.
5. O'Connor,B. Krieger, M., Ahn, D. (2010). TweetMotif: exploratory search and topic summarization for Twitter, in: ICWSM-2010, 2010.
6. Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D. Eisenstein, J., Smith, N. (2010). Part-of-speech tagging for twitter: annotation, features, and experiments, in: Carnegie Mellon Univ. Pittsburgh School of Computer Science, 2010.
7. Owoput, O., O'Connor, B., Dyer, C., Gimpel, K., Schneider, N., Smith, N. (2013). Improved part-of-speech tagging for online conversational text with word clusters, in: HLT-NAACL, 2013, pp. 380–390.
8. Paice, C. (1990). Another stemmer, SIGIR Forum 24 (3) 1990, 56–61.
9. Atkinson, K. (2006). Gnu Aspell 0.60.4, 2006.
10. Porter, M. (2001). Snowball: A Language for Stemming Algorithms, 2001.
11. Toutanova, K., Klein, D., Manning, C., Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. Proceedings of HLT-NAACL 2003, pp. 252–259.
12. Kong, L., Schneider, N., Swayamdipta, S., Bhatia, A., Dyer, C., Smith, N. (2014). A dependency parser for tweets, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, vol. 4, no. 1.2.
13. Pang, Bo; Lee, Lillian (2008). "4.1.2 Subjectivity Detection and Opinion Identification". Opinion Mining and Sentiment Analysis. Now Publishers Inc.
14. Wilson T, Wiebe J, Hoffman P.(2005). Recognizing contextual polarity in phrase-level sentiment analysis. In: Proceedings of HLT/EMNLP; 2005.
15. Liu B. (2012). Sentiment analysis and opinion mining. Synth Lecture Human Language Technology 2012.
16. Turney, Peter (2002). "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews". Proceedings of the Association for Computational Linguistics. pp. 417–424. arXiv:cs.LG/0212032.
17. Pang, Bo; Lee, Lillian (2005). "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales". Proceedings of the Association for Computational Linguistics (ACL). pp. 115–124.

18. Snyder, Benjamin; Barzilay, Regina (2007). "Multiple Aspect Ranking using the Good Grief Algorithm". Proceedings of the Joint Human Language Technology/North American Chapter of the ACL Conference (HLT-NAACL). pp. 300–307.
19. Qu, Yan, James Shanahan, and Janyce Wiebe. "Exploring attitude and affect in text: Theories and applications." In AAAI Spring Symposium) Technical report SS-04-07. AAAI Press, Menlo Park, CA. 2004.
20. Kumar Singh, V., Piryani, R., Waila, P., Devaraj, M. (2013). Computing Sentiment Polarity of Texts at Document and Aspect Levels. ECTI-Con 2013.
21. Zamani, N.A., Abidin, S.Z., Omar, N., & Abiden, M.Z. (2014). Analysis: Determining People ' s Emotions in Facebook.
22. Vinodhini, G., Chandrasekaran, RM. (2012). Sentiment Analysis and Opinion Mining: A Survey. IJAR in Computer Science and Software Engineering, Vol. 2, (6), June 2012.
23. Kaur, A., Gupta, V. (2013). A Survey on Sentiment Analysis and Opinion Mining Techniques. Journal of Emerging Technologies in Web Intelligence, Vol. 5, (4), Nov. 2013
24. Medhat, W., Hassan, A., Korashy, H. (2014). Sentiment analysis algorithms and applications:A survey.
25. Hossein, ED. (2016). A survey on sentiment analysis challenges. Journal of King Saud University – Engineering Sciences 2018 30, 330–338. Ain Shams Engineering Journal (2014) 5, 1093–1113.
26. Ravi, K., Ravi, V. (2015). A survey on opinion mining and sentiment analysis: Tasks, approaches
27. and applications. Knowledge-Based Systems 89 (2015) 14–46.
28. Devika, MD, Sunitha, C., Ganesh, A. (2016). Sentiment Analysis: A Comparative Study on Different Approaches. Procedia Computer Science 87 (2016) 44 – 49.
29. Hinton, G. (2007). "Learning multiple layers of representation". Trends in Cognitive Sciences. **11** (10): 428–434. doi:10.1016/j.tics.2007.09.004. ISSN 1364-6613. P MID 17921042.
30. Hinton, G., Osindero, S., Teh, Y. (2006). "A Fast Learning Algorithm for Deep Belief Nets"(PDF). NeuralComputation. **18** (7):15271554. doi:10.11 62/neco.2006.18.7.1527. PMID 16764513.
31. Szegedy, C., Toshev, A., Erhan, D. (2013). "Deep neural networks for object detection". Advances in Neural Information Processing Systems.
32. Bengio, Y. (2012). "Practical recommendations for gradient-based training of deep architectures". arXiv:1206.5533 [cs.LG].
33. Hinton, G. (2007). "Learning multiple layers of representation," Trends in Cognitive Sciences, 11, pp. 428–434, 2007.
34. Wikipedia, Artificial Neural Network. https://en.wikipedia.org/wiki/Artificial_neural_network
35. Zell, A. (1994). "chapter 5.2". Simulation Neuronaler Netze [Simulation of Neural Networks] (in German) (1st ed.). Addison-Wesley. ISBN 978-3-89319-554-1.
36. Deng, L.; Yu, D. (2014). "Deep Learning: Methods and Applications" (PDF). Foundations and Trends in Signal Processing. 7 (3–4): 1–199. doi:10.1561/2000000039.

Fukushima, K. (1980). "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". Biol. Cybern. 36 (4): 193 202. doi:10.1007/bf00344251. PMID 7370364.

37. Bengio, Y., Courville, A., Vincent, P. (2013). "Representation Learning: A Review and New Perspectives". IEEE Transactions on Pattern Analysis and Machine Intelligence. 35 (8): 1798–1828. arXiv:1206.5538. doi:10.1109/tpami.2013.50.
38. LeCun, Y., Bengio, Y., Hinton, G. (2015). "Deep learning". Nature. 521(7553): 436–444. Bibcode:2015 Natur.521.436L. doi:10.1038/nature14539.
39. "Convolutional Neural Networks (LeNet) – DeepLearning 0.1 documentation". DeepLearning 0.1. LISA Lab. Retrieved 31 August 2013.
40. Habibi, Aghdam, Hamed. Guide to convolutional neural networks : a practical application to traffic-sign detection and classification. Heravi, Elnaz Jahani,. Cham, Switzerland. ISBN 9783319575490. OCLC 987790957.
41. Ciresan, D., Meier, U., Masci, J., Gambardella, L. Schmidhuber, J. (2011). "Flexible, High Performance Convolutional Neural Networks for Image Classification" (PDF). Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Two. 2: 1237–1242.
42. Krizhevsky, A. (2013). "ImageNet Classification with Deep Convolutional Neural Networks" (PDF).
43. Ciresan, D., Meier, U., Schmidhuber, J. (2012). "Multi-column deep neural networks for image classification". 2012 IEEE Conference on Computer Vision and Pattern Recognition. New York, NY: Institute of Electrical and Electronics Engineers(IEEE): 3642–3649. arXiv:1202.2745. doi:10.1109/CVPR.2012.6248110. ISBN 978-1-4673-1226-4. OCLC 812295155.
44. Cícero Dos Santos., Gatti, M. (2014). Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts. In COLING. 69–78.
45. Kalchbrenner, N., Grefenstette, E., Blunsom, P. (2014). A convolutional neural network for modelling sentences. arXiv preprint arXiv:1404.2188.
46. Kim, Y. (2014). Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014).

Zeng, D., Liu, K., Lai, S., Zhou, G., Zhao, J. (2014). Relation Classification via Convolutional Deep Neural Network. In COLING. 2335–2344.

47. Mittal, S. (2018)."A Survey of FPGA-based Accelerators for Convolutional Neural Networks", Academia, NCAA, 2018
48. Graves, A.; Liwicki, M.; Fernandez, S.; Bertolami, R.; Bunke, H.; Schmidhuber, J. (2009). "A Novel Connectionist System for Improved Unconstrained Handwriting Recognition" (PDF). IEEE Transactions on Pattern Analysis and Machine Intelligence. 31 (5). doi:10.1109/tpami.2008.137.
49. [Sak, Hasim; Senior, Andrew; Beaufays, Francoise (2014). "Long Short-Term Memory recurrent neural network architectures for lge scale acoustic modeling" (PDF).
50. Li, Xiangang; Wu, Xihong (2014). "Constructing Long Short-Term Memory based Deep Recurrent Neural Networks for Large Vocabulary Speech Recognition". arXiv:1410.4281 [cs.CL].
51. Goller, C.; Küchler, A. "Learning task-dependent distributed representations by backpropagation through structure". Neural Networks, 1996., IEEE. doi:10.1109/ICNN.1996.548916.
52. Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C., Ng, A., Potts, C. (2013). “Recursive deep models for semantic compositionality over a sentiment treebank,” in Proceedings of the conference on empirical methods in natural language processing (EMNLP), vol. 1631, p. 1642.
53. Pak, A.,Paroubek, P. (2010). Twitter as a Corpus for Sentiment Analysis and Opinion Mining Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, 17-23, Valletta, Malta.
54. Pham, V., Bluche, T., Kermorvant, C., Louradour, J.(2013). Dropout improves Recurrent Neural
55. Networks for Handwriting Recognition. arXiv:1312.4569 [cs].URL http://arxiv.org/abs/1312.4569.
56. Doetsch, P., Kozielski, M., Ney, H. (2014). Fast and robust training of recurrent neural networks foroffline handwriting recognition. In 14th International Conference on Frontiers in Handwriting Recognition. URL http://people.sabanciuniv.edu/berrin/cs581/ Papers/icfhr2014/data/4334a279.pdf
57. Graves, A. (2013). Generating sequences with recurrent neural networks. arXiv:1308.0850 [cs] , August 2013. URL http://arxiv.org/abs/1308.0850

Zaremba, W., Sutskever, I., Vinyals, O. (2014). Recurrent Neural Network Regularization. arXiv:1409.2329 [cs], September 2014. URL http://arxiv.org/abs/1409.2329

58. Luong, T., Sutskever, I., Le, Q., Vinyals, O, Zaremba, W. (2014). Addressing the Rare Word Problem in Neural Machine Translation. arXiv preprint arXiv:1410.8206. URL http://arxiv.org/abs/1410.8206.

59. Sak, H., Senior, A., Beaufays, F.(2014) Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH). URL http://193.6.4.39/czap/letoltes/IS14/IS2014/PDF/AUTHOR/IS141304.PDF.
60. Fan, Y., Qian, Y., Xie, F., Soong, F.(2014).TTS synthesis with bidirectional LSTM based recurrent neural networks. In Proc. Interspeech.
61. Sønderby, SK., Winther, O. (2014). Protein Secondary Structure Prediction with Long Short-Term Memory Networks. arXiv:1412.7828 [cs, q-bio], December 2014. URL http://arxiv.org/abs/1412.7828.arXiv: 1412.7828.
62. Marchi, E., Ferroni, G., Eyben, F., Gabrielli, L., Squartini, S., Schuller, B.(2014). Multi-resolution linear prediction based features for audio onset detection with bidirectional LSTM neural networks. In 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),pages 2164–2168. doi: 10.1109/ICASSP.2014.6853982.
63. Donahue,J., Hendricks, LA, Guadarrama, S, Rohrbach, M., Venugopalan, S., Saenko, K. (2014) and Trevor Darrell. Long-term Recurrent Convolutional Networks for Visual Recognition and Description. arXiv:1411.4389 [cs]. URL http://arxiv.org/abs/1411.4389. arXiv: 1411.4389.
64. Domingos, P.(2015). The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World. Basic Books. "Deeper into the Brain" subsection.
65. Kingma, D., Welling, M. (2013). "Auto-Encoding Variational Bayes". arXiv:1312.6114[stat.ML].
66. Hutson, M. (2018). "New algorithm can create movies from just a few snippets of text". Science. doi:10.1126/science. aat4126.
67. Vincent, P. Larochelle, H., Lajoie, I. Bengio, Y., Manzagol, PA. (2010). "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion". The Journal of Machine Learning Research. **11**: 3371–3408.
68. Zhao, L., Hu, Q., Wang, W., (2015). Heterogeneous Feature Selection with Multi-Modal Deep Neural Networks and Sparse Group LASSO. IEEE Transactions on Multimedia, Vol. 17, (11), November 2015.
69. Makhzani, A. Frey, B. (2013). "k-sparse autoencoder". arXiv:1312.5663 [cs.LG].
70. Wan, X., Xiao, J. (2008). Single Document Keyphrase Extraction using Neighborhood Knowledge. In: Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence.
71. Shen, D., Sun, J.T., Li, H., Yang, Q., Chen, Z.(2007). Document Summarization using Conditional Random Fields. In: Proceedings of the 20th international joint conference on Artificial Intelligence, pp. 2862-2867. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.
72. Wong, K.F., Wu, M.J., Li, W.J.(2008). Extractive Summarization Using Supervised and Semi-Supervised Learning. In: Proceedings of the 22nd International Conference on Computational Linguistics Volume 1, pp. 985-992. Association for Computational Linguistics Stroudsburg, PA, USA.
73. Jin, F., Huang, M.L., and Zhu, X.Y. (2010). A Comparative Study on Ranking and Selection Strategies for Multi Document Summarization. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters, pp. 525-533. Association for Computational Linguistics Stroudsburg, PA, USA.
74. Esser, D., Muthmann, K., Schsuter, D., Schill, A., (2012). Automatic Indexing of Scanned Documents - a Layout-based Approach. Proceedings of SPIE - The International Society for Optical Engineering 8297 · January 2012. DOI: 10.1117/12.908542
75. Appelt, D., Israel, D. (1999). Introduction to Information Extraction Technology A Tutorial Prepared for IJCAI-99.
76. [71] Cunningham, H., Maynard, D., Bontcheva, K., Tablan. V (2002). GATE: A framework and graphical development environment for robust NLP tools and applications. In Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02), Philadelphia.
77. Costantino, M., Coletti, P. (2008). Information Extraction in Finance, Wit Press, 2008. ISBN 978-1-84564-146-7
78. Hobbs, J., Appelt, D., Tyson, M., Bear, J., Islael, D. (1992). SRI International: Description of the FASTUS system used for MUC-4. In Proceedings fo the 4th Message Understanding Conference (MUC-4), pages 268–275.
79. Krupka, G., Jacobs, P., Rau, L., Childs, L., Sider, I (1992). GE NLTOOLSET: Description of the system as used for MUC-4. In Proceedings of the 4th Message Understanding Conference (MUC-4), pages 177–185.
80. Ayuso, D., Boisen, S., Fox, H., Gish, H., Ingria, R., Weischedel, R. (1992). BBN: Description of the PLUM system as used for MUC-4. In Proceedings of the Fourth Message Understanding Conference (MUC-4), pages 169–176.
81. Riloff, E. (1993). Automatically constructing a dictionary for information extraction tasks. In Proc. of the 11th National conference on Artificial Intelligence, pages 811–816.
82. Kim, J., Moldovan, D. (1995). Acquisition of linguistic patterns for knowledge-based information extraction. IEEE Transactions on Knowledge and Data Engineering, 7(5):713–724.
83. Soderland, S., Fisher, D., Aseltine, Lehnert, W. (1995). CRYSTAL: inducing a conceptual dictionary. In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI'95), pages 1314–1319.
84. Huffman., S. (1996), Learning information extraction patterns from examples. In Lecture Notes in Computer Science. Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing, volume 1040, pages 246–260, London, UK, Springer Verlag.
85. Riloff, E. (1996). Automatically generating extraction patterns from untagged text. In Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96), pages 1044–1049.
86. Riloff, E., and Jones, R. (1999). Learning dictionaries for information extraction by multilevel bootstrapping. In Proceedings of the 16th National Conference on Artificial Intelligence, pages 474–479. AAAI Press/MIT Press.
87. Yangarber, R., Grishman, R., Tapanainen, P., Huttunen, S. (2000). Automatic acquisition of domain knowledge for information extraction. In Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000), Saarbr¨ucken, Germany, August 2000.
88. Agichtein, E., and Gravano, L. (2000). Snowball: Extracting relations from large plaintext collections. In Proceedings of the 5th ACM International Conference on Digital Libraries.
89. Brin., S. (1998). Extracting patterns and relations from the world wide web. In WebDB Workshop at 6th International Conference on Extended Database Technology, EDBT'98.
90. Sudo, K. (2004). Unsupervised Discovery of Extraction Patterns for Information Extraction. PhD thesis, New York University, New York.
91. Sudo, K., Sekine, S., and Grishman, R. (2001). Automatic pattern acquisition for Japanese Information Extraction. In Proceedings of Human Language Technology Conference (HLT2001), San Diego, CA.
92. Young, T., Hazarika, D., Poria, S., Cambria, E. (2018). Recent Trends in Deep Learning Based Natural Language Processing.arXiv:1708.02709v7 [cs.CL]
93. Mikolov, T., Karafi´at, M., Burget, L., Cernock`y, J., Khudanpur, S. (2010). Recurrent neural network based language model. Interspeech, vol. 2, p. 3.
94. Mikolov, T., Karafi´at, M., Burget, L., Cernock`y, J., Khdanpur, S. (2011). Extensions of recurrent neural network language model. Acoustics, Speech and Signal Processing (ICASSP), International Conference, IEEE,pp. 5528–5531.
95. Sutskever, I., Martens, J., Hinton, G.(2011). Generating text with recurrent neural networks. Proceedings of the 28th International Conference on Machine Learning (ICML-11), pp. 1017–1024.
96. Liu, S., Yang, N., Li, M., Zhou, M. (2014). A recursive recurrent neural network for statistical machine translation. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pp. 1491–1500.

97. Auli, M., Galley, M., Quirk, C., Zweig, G. (2013). Joint language and translation modeling with recurrent neural networks. EMNLP, 2013, pp. 1044–1054.
98. Sutskever, I., Vinyals, O., Le, Q. (2014). Sequence to sequence learning with neural networks. Advances in neural information processing systems, pp. 3104–3112.
99. Kalchbrenner, N., Grefenstette, E., Blunsom, P. (2014). "A convolutional neural network for modelling sentences," Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. [Online].Available: http://goo.gl/EsQCuC
100. Kim, Y. (2014). Convolutional neural networks for sentence classification.arXivpreprint arXiv:1408.5882.
101. [96] Robinson, T., Hochberg, M., Renals, S. (1996). The use of recurrent neural networks in continuous speech recognition. Automatic speech and speaker recognition. Springer. pp. 233–258.
102. Graves, A., Mohamed, A., Hinton, G. (2104). Speech recognition with deep recurrent neural networks. Acoustics,speech and signal processing (ICASSP-14), international conference on IEEE, pp. 6645–6649.
103. Graves A., Jaitly, N. (2014). Towards end-to-end speech recognition with recurrent neural networks. Proceedings of the 31st International Conference on Machine Learning (ICML-14), pp. 1764–1772.
104. Sak, H., A. Senior, A., Beaufays, F.(2014). Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition.arXiv preprint arXiv:1402.1128.
105. Karpathy, A. Fei-Fei, L.(2015). Deep visual-semantic alignments for generating image descriptions. Proceedings of theIEEE Conference on Computer Vision and Pattern Recognition, pp. 3128–3137.
106. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. International Conference on Machine Learning. 2048–2057.
107. Yang, Y., Yih, W., Meek, C.(2015). Wikiqa: A challenge dataset for open-domain question answering. International Conference on Empirical Methods in Natural Language Processing. 2013–2018.
Cho, K., Merriënboer, B., Bahdanau, D., Benigo, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259.
108. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.(2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.
109. Zaremba, W., Sutskever, I., Vinyals, O. (2014). Recurrent neural network regularization. arXiv preprint arXiv:1409.2329.
110. Bowman, S., Vilnis, L., Vinyals, O., Dai, A., Jozefowicz, R., Bengio, Y. (2015). Generating sentences from a continuous space. arXiv preprint arXiv:1511.06349.
111. Hu, Z., Yang, Z., Liang, X., Salakhutdinov, R., Xing, E. (2017). Controllable text generation. arXiv preprint arXiv:1703.00955.
112. Srinivas V., Gelgi, F., Hasan, D. (2007). Information Extraction from Web Pages using Presentation Regularities and Domain Knowledge. Journal of World Wide Web, Springer Netherlands, Arizona State University, USA, Vol. 10, Issue 2, pp. 157-179.
113. Lam, M., and Gong, Z. (2005). Web Information Extraction, Proceedings of IEEE International Conference on Information Acquisition, pp. 6.
114. Horacio Saggion, H., Funk, A., Maynard, D., Bontcheva K. (2008). Ontology based Information Extraction for Business Intelligence, In: Lecture Notes in Computer Science, pp. 843-856.
115. Rahman, K. (2017). Aye Aye AI - Golden Age of Innovation in Artificial Intelligence & Computer Science – How will this impact Humans? Accessed on Nov. 5, 2018 from https://ionmedicalsafety.org/blog/aye-aye-ai-golden-age-of-innovation-in-artificial-intelligence-computer-science-how-will-this-impact-humans
116. Gill N.S. (2017). Overview of Artificial Intelligence, Deep Learning and NLP in Big Data. Accessed on Nov. 5, 2018 from https://www.xenonstack.com/blog/data-science/ai-nlp-big-deep-learning/
117. Wikipedia. Artificial Nueral Network, Media file. Accessed on Nov. 5, 2018 from https://en.wikipedia.org/wiki/Artificial_neural_network#/media/File:Neuron3.png
118. Wikipedia. Convolutional Neural Network. Media file. Accessed on Nov. 5, 2018 from https://en.wikipedia.org/wiki/Convolutional_neural_network#/media/File:Typical_cnn.png
119. Wikipedia. Recurrent Neural Network. Media file. Accessed on Nov. 5, 2018 from https://en.wikipedia.org/wiki/Recurrent_neural_network#/media/File:Recurrent_neural_network_unfold.svg
120. Wikipedia. Long Short-Term Memory (LSTM). Media file. Accessed on Nov. 5, 2018 from https://en.wikipedia.org/wiki/Long_short-term_memory#/media/File:The_LSTM_cell.png
121. Wikipedia. Auto encoder. Media file. Accessed on Nov. 5, 2018 from https://en.wikipedia.org/wiki/Autoencoder#/media/File:Autoencoder_structure.png
122. Frans, K. (2016). Variational Autoencoders Explained. Accessed on Nov. 5, 2018 from http://kvfrans.com/variational-autoencoders-explained/

AUTHORS

1. **Joseph M. De Guia**

Josephis a PhD student from Mapua University, Manila Philippines. He is currently afaculty member of the School of Information Technology of same university. Joseph finished his masters degree in Information Technology (MSIT) at Carnegie Mellon University and in Computer Science (MSCS) at Mapua University. His research interests are data mining, artificial intelligence, information security, big data analytics, IT infrastructure and digital innovation. Hisresearch papers in digital health and enterprise architecture has been presented in international and local conferences.

2. **Madhavi Devaraj**

Dr. Devaraj received doctoral degree in computer science from Dr. A.P.J Abdul Kalam Technical University, Lucknow, India. She has also completed Masters in Computer Applications andMPhil in Computer Science from Madurai Kamaraj University, Madurai, India. Currently, she is distinguished professor in computer science department at Mapua University, Manila, Philippines.She has been assistant professor in computer science department at Invert is University, India and Babu Banarasi Das University, India, previously. Her research interests include Text analytics,Scientometric Analysis, Opinion Mining, Sentiment Analysis, Information Extraction, Neural Networks, Artificial Intelligence, Machine Learning and Big Data Analysis.